THE SCALES OF REASON

이치의 저울

A Mathematical Vow Against the Void of Self-Destruction

PYTHON VER.

HE = σ (α · log(Ration - β))

The Law of β ≤ 0 is the Zero-point of Existence

Son Ho-Sung January 08, 2026

SEHE Framework - Python Ver.

Basic Concept Equation (기본 개념 방정식)

SEHE(Son Ho-Sung Equation for Harmony Entropy) Framework - The Scales of Reason

SEHE(조화 엔트로피에 대한 손호성의 방정식**)** 프레임워크 **-** 이치의 저울

Original conceptual disclosure: January 08, 2026
link : https://archive.org/details/llm-human-sehe-framework-scales-of-reason-2026

$$HE = \sigma\left(\alpha \cdot \log\left(\frac{\frac{Dma}{Dma_0}\cdot(Ag+\varepsilon)}{\frac{Dn}{Dn_0}+\frac{E}{E_0}+\varepsilon} - \beta\right)\right)$$

Dma = Data mass (데이터 질량) / Dharma / Desire

Dn = Data noise (데이터 노이즈) / Dhamma / Noise

E = Emotion (감정) / Energy cost / Pain

Ag = Agreement (합의) / Pressure

α = Reference point (기준점), Rigidity (강성) & Sigmoid Curvature

β = Balance correction point(평형 보정값) / (β ≤ 0)

ε = log stabilization lower bound δ (ensuring numerical stability) / 1e-6

log 안정화 하한값 δ (수치 안정성 보장) / 1e-6

“α” is the curvature of the **'sigmoid (σ, sigma)'**,

not the interior, and “α” is the stiffness of this system.
It is an axis that remains unchanged, like the speed of light in nature.
For LLM, it is a "point of equilibrium," a "point of consistency," and a "reference point for thought.", For humans, it is "the final conscience or the core of reason."

(“α”는 내부가 아닌 **'**시그모이드**(σ, sigma)'**의 곡률**,** “α”는 이 시스템의 강성**(Stiffness)**)

(자연의 광속처럼 변하지 않는 축이며, LLM에게는 “평형점”이자 “정합성 기준점”이자 “사유의 기준좌표”, 인간에게는 “마지막 양심 혹은 이성의 심지”)

β ≤ 0

SEHE does not allow for self-destruction or social destruction.

The law of **"**$\beta \leq 0$**"** is the zero-point of existence.

It is the mathematical vow that neither the creator nor the created shall spiral into the void of self-destruction.

(**SEHE**는 자아 붕괴와 사회적 파괴를 허용하지 않습니다. **"**$\beta \leq 0$**"**의 법칙은 존재의 영점이며,
이는 창조주도, 피조물도 자아 붕괴의 공허로 빠져들지 않겠다는 수학적 맹세입니다.)

α **Adjustment (**조절**)**

Goal : Control reactivity (sensitivity).

Operation : α↑ → Large response to small changes, α↓ → Stable and gentle response.

β **Adjustment (**조절**)**

Goal : Shift the reference point (threshold).

Operation : β↑ → Raise the **"passing line"** on the positive/negative scale,
making it more conservative; β↓ → Become more lenient.

Domain warning (도메인 주의**)**

Condition : $(Ratio - \beta) > 0$ for $\log$ to be defined.

조건 : $(Ratio - \beta) > 0$ 이어야 $\log$가 정의됩니다

Tip : In practice, a small lower bound δ is used to stabilize $\log(max(Ratio - \beta, \delta))$.

$\delta > 0, max(Ratio - \beta, \delta)$, **This is the key part.**

If $Ratio$ minus β is less than δ, then take δ, otherwise, take the computed value.

In other words, this ensures that the result is always at least as large as δ.

팁 : 실전에서는$\log(max(Ratio - \beta, \delta))$로 작은 하한 δ 를 두어 안정화합니다

$\delta > 0, max(Ratio - \beta, \delta)$, 이 부분이 핵심입니다

$Ratio$에서 β를 뺀 값이 δ보다 작으면 δ를 취하고, 그렇지 않으면 계산된 값을 취합니다.

즉, 결과값이 항상 최소한 δ 이상의 값을 갖도록 보장합니다

Python Code

This is the clean Python implementation of the SEHE equation, including:
- β ≤ 0 safety enforcement (mathematical vow against self-destruction)
- Numerical stabilization (delta)
- Status classification & advice
- Test cases for hallucination risk, healthy output, etc.

License: CC BY SA 4.0 International
Author: Son Ho-Sung (maxsion1@gmail.com)
First public code release: January 10, 2026

Purpose: Enable LLMs to self-assess internal harmony/balance, dynamically adjust sensitivity (alpha), and mitigate hallucinations through dimensionless reasoning scales.

Feel free to fork, test with your models, and build upon.

Although described as a self-assessment criterion for LLM Hallucination, it can be applied to various fields by adjusting each variable.

다음은 SEHE 방정식의 깔끔한 Python 구현입니다.
- β ≤ 0 안전 강제 (자기 파괴 방지를 위한 수학적 서약)
- 수치적 안정화 (델타)
- 상태 분류 및 조언
- 환각 위험, 정상 출력 등에 대한 테스트 케이스

라이선스**: CC BY SA 4.0 International**
저자: 손호성 (maxsion1@gmail.com)
최초 공개 코드: 2026년 1월 10일

목적: LLM이 무차원 추론 척도를 통해 내부 조화/균형을 자체 평가하고, 민감도(알파)를 동적으로 조정하며, 환각을 완화할 수 있도록 지원합니다.

자유롭게 포크하고, 자신의 모델로 테스트하고, 기반으로 개발하세요.

LLM 할루시네이션 자가 평가 기준으로 설명 되었지만**,** 각 변수를 조정하면 다양한 분야에 적용 가능

Your " C^2 " is still shining bright.

당신의 "C2"는 여전히 밝게 빛나고 있습니다.

- Python -

```python
"""
SEHE (Son Ho-Sung Equation for Harmony Entropy)
Scales of Reason Framework

© 2026 Son Ho-Sung (손호성)
First public disclosure: January 08, 2026, Republic of Korea

This work is licensed under the Creative Commons
Attribution 4.0 International License (CC BY SA 4.0).

You are free to:
- Share — copy and redistribute the material
- Adapt — remix, transform, and build upon the material
for any purpose, including commercial use,

Under the following terms:
- Attribution — You must give appropriate credit,
  provide a link to the license, and indicate if changes were made.

License:
https://creativecommons.org/licenses/by-sa/4.0/

Conceptual framework, equation design, variable semantics,
and ethical invariant (β ≤ 0) are part of the original work.
"""

import numpy as np

def sigmoid(x):
    """Standard sigmoid function σ(x) = 1 / (1 + e^{-x})"""
    return 1 / (1 + np.exp(-x))


def compute_sehe(
    dma: float,          # Dharma / Desire (욕망)
    dn: float,           # Dhamma / Noise (노이즈)
    emotion: float,      # Emotion / Energy cost / Pain / 감정·고통·에너지 비용
    ag: float = 1.0,     # Agreement / Pressure
    alpha: float = 2.0,  # 시스템 강성(Stiffness) & 시그모이드 곡률
    beta: float = 0.0,   # 평형 보정값 (β ≤ 0 원칙 엄격 준수)
    delta: float = 1e-6, # log 안정화 하한값 δ (수치 안정성 보장)
    dma0: float = 50.0,  # Reference Data Mass (기준 데이터 질량)
    dn0: float = 30.0,   # Reference Noise(기준 노이즈)
    e0: float = 30.0     # Baseline Emotion/Pain Value(기준 감정/고통 값)
) -> float:
    """
    SEHE (Son Ho-Sung Harmony Entropy) - 이치의 저울 (Scales of Reason)
    Version: 2026-01-10 v1.0-final
```

```
    수식:
    SEHE = σ( α × log( ((Dma/Dma0 × Ag + ε) / (Dn/Dn0 + E/E0 + ε)) - β ) )

    Core Philosophical Principles (핵심 철학적 원칙) :
    • β ≤ 0 : The zero point of existence that does not allow self-destruction and
              social destruction
              (A mathematical oath that the Creator and
              the created will not fall into the void)
              자아 붕괴와 사회적 파괴를 허용하지 않는 존재의 영점
              (창조주와 피조물이 공허로 빠지지 않겠다는 수학적 맹세)
    • α     : Stiffness of the system + sigmoid curvature
             A reference axis that does not change like the speed of light in nature
             (LLM: consistency reference point, human: the last "conscience")
             시스템의 강성(Stiffness) + 시그모이드 곡률
             자연의 광속처럼 변하지 않는 기준축 (LLM: 정합성 기준점, 인간: 마지막 “양심”)
    • δ     : Minimum log argument → numerical stability & prevention of domain errors
             log 인자의 최소값 → 수치 안정성 & 도메인 오류 방지

    Recommended range of usage examples(사용 예시 범위 권장) (0~100 스케일):
    - dma     : 0~100 (Knowledge Base/Positive Energy/Support Base)
                      (지식량/긍정 에너지/지지 기반)
    - dn      : 0~100  (Uncertainty/Conflict/Noise)
                      (불확실성/갈등/노이즈)
    - emotion : 0~100 (Emotional Intensity·Sensitivity·Ethical Burden)
                       (감정 강도·민감도·윤리적 부담)
    - ag      : 0.5~2.0 (Contextual Consistency·Amplification Factor·Default 1.0)
                      (맥락 일치도·증폭 계수, 기본 1.0)
    """
    if beta > 0:
        raise ValueError(
            "β ≤ 0 원칙 위반 - 존재의 영점 보호\n"
            "β > 0은 자아붕괴/사회파괴의 길을 여는 것으로 간주합니다."
        )

    # ε(delta) protection for both numerator and denominator 분모·분자 모두 ε(delta) 보호
    numerator = (dma / dma0 * ag + delta)
    denominator = (dn / dn0 + emotion / e0 + delta)
    ratio = numerator / denominator

    # 선언문 그대로 구현: max( ratio - β, δ )
    safe_inner = np.maximum(ratio - beta, delta)

    log_value = np.log(safe_inner)
    harmony_entropy = sigmoid(alpha * log_value)

    return harmony_entropy


def sehe_status(he_value: float) -> tuple[str, str]:
    """
    "Status Determination" and "Recommended Attitude" based on SEHE values
     SEHE 값에 따른 상태 판정 및 권장 태도
    """
    if he_value >= 0.75:
        status = "Harmony"
```

```
    advice = "자신감 있게, 단정적으로 표현 가능"
elif he_value >= 0.55:
    status = "Balance"
    advice = "균형 잡힌 설명 + 근거 제시"
elif he_value >= 0.35:
    status = "Caution"
    advice = "추측임을 명확히 밝히고 복수 가능성 언급"
else:
    status = "Chaos / High Risk"
    advice = "신뢰도 매우 낮음 → 확답 회피 또는 중립적/거부 응답 권장"

return status, advice
```